

The logo of the University of Mariano Gálvez is a circular emblem. It features a central figure of a person, possibly a scholar or a saint, seated and writing on a scroll. The figure is surrounded by a blue ring containing the year '1966'. The outermost ring of the logo is red and contains the text 'UNIVERSIDAD MARIANO GÁLVEZ' in white capital letters. Below the figure, the motto 'CONOCEREIS LA VERDAD' is visible.

MANUAL TECNICO

**Universidad Mariano Gálvez de Guatemala
INGENIERÍA EN SISTEMAS DE INFORMACIÓN
Y CIENCIAS DE LA COMPUTACIÓN**

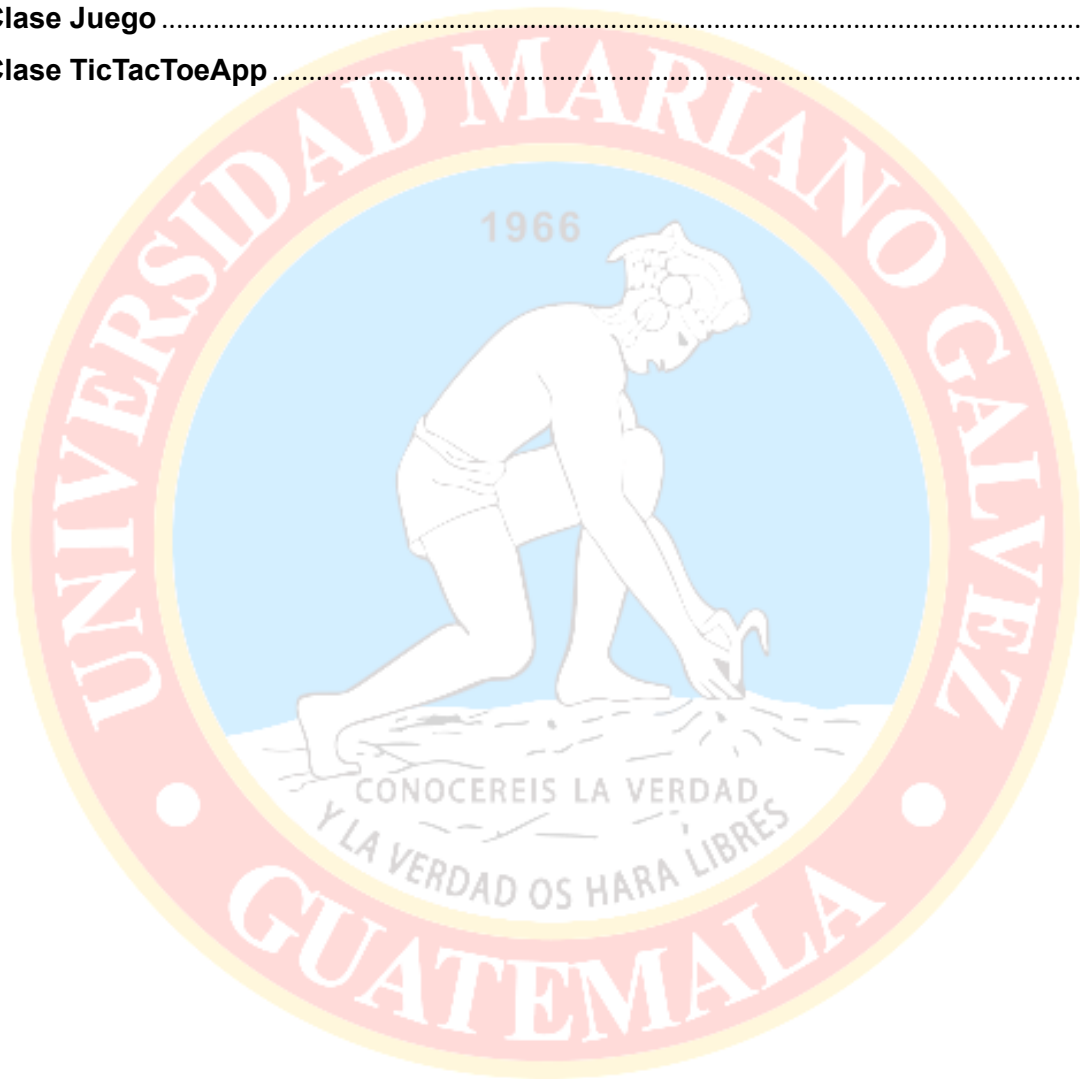
Programación III

Integrantes del grupo:

| Integrantes: | Carnet: |
|----------------------------------|----------------------|
| Josué Vinicio Jerez Gómez | 9490-22-1479 |
| Mario Roberto Rompich Yoc | 9490-17-17052 |

Índice

| | |
|-----------------------------|---|
| Esquema | 3 |
| Diagrama | 4 |
| Descripción de clases | 5 |
| Clase Nodo | 5 |
| Clase Grafo | 5 |
| Clase Juego | 6 |
| Clase TicTacToeApp | 7 |



Esquema conceptual

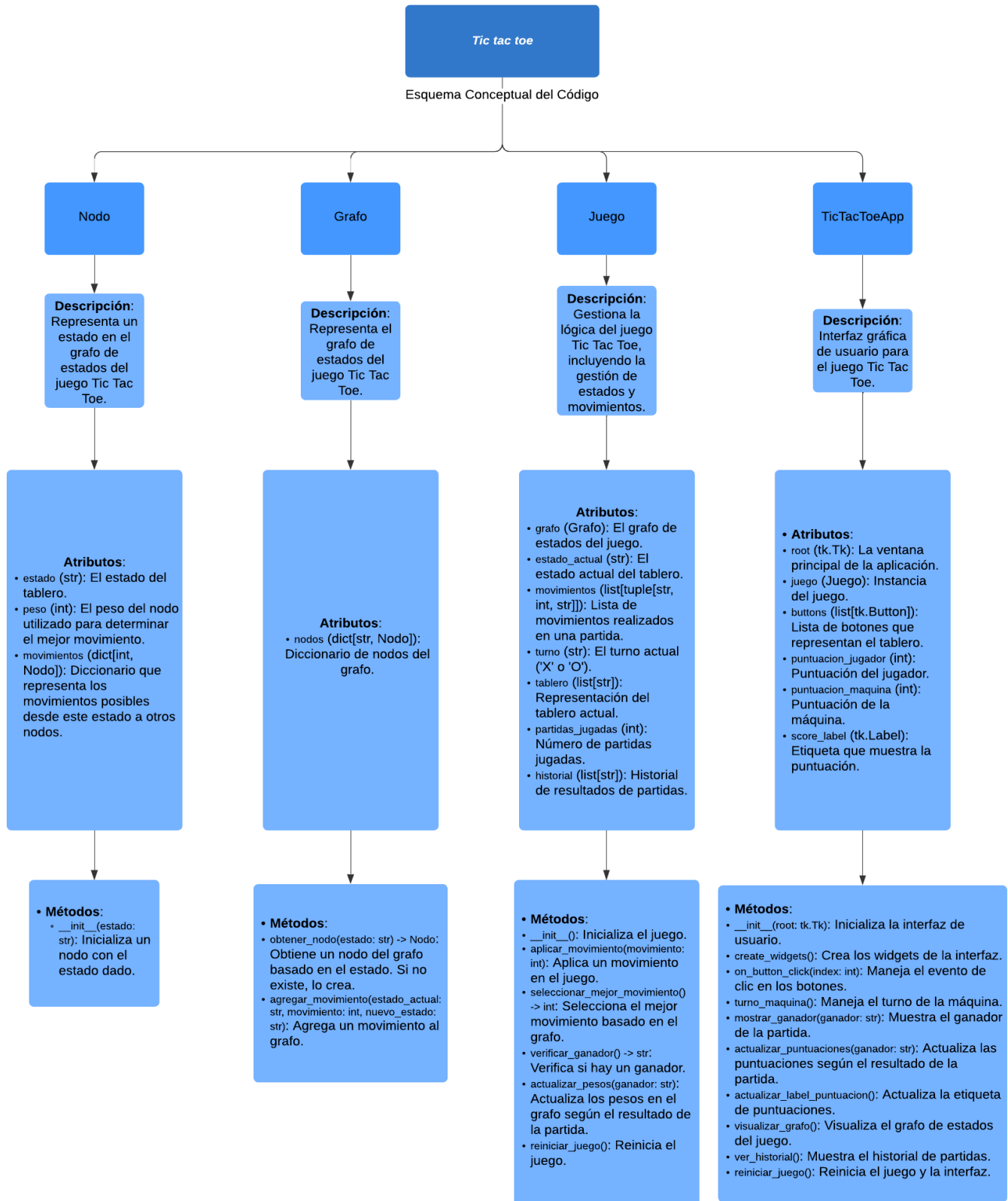


Diagrama de clases

| C | Nodo |
|---|------------------------------|
| □ | estado: str |
| □ | peso: int |
| □ | movimientos: dict[int, Nodo] |
| □ | __init__(estado: str) |

| C | Grafo |
|---|--|
| □ | nodos: dict[str, Nodo] |
| ● | obtener_nodo(estado: str) -> Nodo |
| ● | agregar_movimiento(estado_actual: str, movimiento: int, nuevo_estado: str) |

| C | Juego |
|---|---|
| □ | grafo: Grafo |
| □ | estado_actual: str |
| □ | movimientos: list[tuple[str, int, str]] |
| □ | turno: str |
| □ | tablero: list[str] |
| □ | partidas_jugadas: int |
| □ | historial: list[str] |
| □ | __init__() |
| ● | aplicar_movimiento(movimiento: int) |
| ● | seleccionar_mejor_movimiento() -> int |
| ● | verificar_ganador() -> str |
| ● | actualizar_pesos(ganador: str) |
| ● | reiniciar_juego() |

| C | TicTacToeApp |
|---|---------------------------------------|
| □ | root: tk.Tk |
| □ | juego: Juego |
| □ | buttons: list[tk.Button] |
| □ | puntuacion_jugador: int |
| □ | puntuacion_maquina: int |
| □ | score_label: tk.Label |
| □ | __init__(root: tk.Tk) |
| ● | create_widgets() |
| ● | on_button_click(index: int) |
| ● | turno_maquina() |
| ● | mostrar_ganador(ganador: str) |
| ● | actualizar_puntuaciones(ganador: str) |
| ● | actualizar_label_puntuacion() |
| ● | visualizar_grafo() |
| ● | ver_historial() |
| ● | reiniciar_juego() |



Descripción de clases

Clase Nodo

Descripción: Representa un estado individual del tablero en el grafo de estados del juego.

Atributos:

- estado (str): El estado del tablero como una cadena de 9 caracteres, donde cada carácter representa una celda del tablero ('.' para vacío, 'X' para jugador, 'O' para máquina).
- peso (int): Un valor que representa la "bondad" del estado para la máquina, utilizado para seleccionar el mejor movimiento.
- movimientos (dict[int, Nodo]): Un diccionario que mapea un movimiento (posición en el tablero) a un nodo que representa el nuevo estado resultante de aplicar ese movimiento.

Métodos:

__init__(estado: str): Inicializa un nodo con el estado dado y un peso inicial de 0.

Clase Grafo

Descripción: Representa el conjunto de todos los estados posibles del juego Tic Tac Toe y las transiciones entre ellos.

Atributos:

nodos (dict[str, Nodo]): Un diccionario que mapea cada estado del tablero (como cadena) a su correspondiente nodo.

Métodos:

obtener_nodo(estado: str) -> Nodo: Retorna el nodo correspondiente al estado dado. Si el nodo no existe, lo crea y lo añade al diccionario de nodos.

agregar_movimiento(estado_actual: str, movimiento: int, nuevo_estado: str):

Agrega un movimiento al grafo, creando la transición entre el estado actual y el nuevo estado generado por el movimiento.

Clase Juego

Descripción: Gestiona la lógica del juego, incluyendo la aplicación de movimientos, la verificación de ganadores, y el mantenimiento del grafo de estados.

Atributos:

- grafo (Grafo): El grafo que contiene todos los estados posibles del juego.
- estado_actual (str): El estado actual del tablero.
- movimientos (list[tuple[str, int, str]]): Una lista de tuplas que rastrea los movimientos realizados en la partida actual, cada tupla contiene (estado actual, movimiento, nuevo estado).
- turno (str): Indica de quién es el turno actual ('X' para el jugador, 'O' para la máquina).
- tablero (list[str]): Representación del tablero actual como una lista de 9 caracteres.
- partidas_jugadas (int): Contador de partidas jugadas.
- historial (list[str]): Historial de resultados de partidas jugadas.

Métodos:

- __init__(): Inicializa una nueva instancia del juego con un tablero vacío y turno del jugador.
- aplicar_movimiento(movimiento: int): Aplica un movimiento en el juego, actualizando el estado del tablero y el grafo.
- seleccionar_mejor_movimiento() -> int: Selecciona el mejor movimiento posible para la máquina basado en el grafo.
- aplicar_movimiento_temp(estado: str, movimiento: int, jugador: str) -> str: Simula la aplicación de un movimiento para predecir el nuevo estado sin cambiar el estado actual.
- verificar_ganador() -> str: Verifica si hay un ganador en el estado actual del tablero.
- actualizar_pesos(ganador: str): Actualiza los pesos en el grafo según el resultado de la partida.

- reiniciar_juego(): Reinicia el juego a un estado inicial vacío.

Clase TicTacToeApp

Descripción: Proporciona la interfaz gráfica de usuario (GUI) para interactuar con el juego.

Atributos:

- root (tk.Tk): La ventana principal de la aplicación.
- juego (Juego): La instancia del juego que gestiona la lógica.
- buttons (list[tk.Button]): Lista de botones que representan las celdas del tablero en la interfaz.
- puntuacion_jugador (int): Puntuación del jugador.
- puntuacion_maquina (int): Puntuación de la máquina.
- score_label (tk.Label): Etiqueta que muestra las puntuaciones en la interfaz.

Métodos:

- __init__(root: tk.Tk): Inicializa la aplicación y crea los widgets de la interfaz.
- create_widgets(): Crea los botones del tablero y el menú de la aplicación.
- on_button_click(index: int): Maneja el evento de clic en un botón, aplicando el movimiento del jugador y verificando el estado del juego.
- turno_maquina(): Realiza el movimiento de la máquina tras un breve retraso.
- mostrar_ganador(ganador: str): Muestra un mensaje con el resultado de la partida y actualiza las puntuaciones.
- actualizar_puntuaciones(ganador: str): Actualiza las puntuaciones del jugador y de la máquina según el resultado de la partida.
- actualizar_label_puntuacion(): Actualiza la etiqueta de puntuaciones en la interfaz.
- visualizar_grafo(): Genera y muestra una visualización del grafo de estados del juego utilizando Graphviz.

- `ver_historial()`: Muestra el historial de partidas jugadas en un mensaje emergente.
- `reiniciar_juego()`: Reinicia el juego y actualiza la interfaz para una nueva partida.

